

**REPLACEMENT SPECIFICATION SHOWING CHANGES****Secure Usage of Digital Certificates
and Related Keys on a Security Token****Background of the invention**

The present invention relates to a security token and method for secure usage of digital certificates and related keys on a security token, and more particularly, to a secure import of certificates into a security token and their secure usage by applications.

If a customer orders some goods or services, he often has to sign a contract on paper to testify that he placed the order and is liable to pay for it. If the customer makes the deal over an electronic network instead, he needs the electronic equivalent of signing a paper: a digital signature. Such a digital signature must guarantee that a customer cannot repudiate his order.

The different methods for creating a digital signature are based on an asymmetrical key pair. The signing person has a private key ~~which~~ that cannot be accessed or used by anybody else. A second key, which is associated ~~to~~ with the private key, is known to the public. This key is called a public key. Only the unique owner of the private key can sign an order, while everybody can check the signature using the corresponding public key.

The public key is distributed in a certificate, which contains the owner's name and a public key and some further information. In addition, the certificate has an expiration date. A reasonable question is, "how do we know that the public key in the certificate is not manipulated?" The answer is that a trusted authority digitally signed the certificate. To check the certificate

signature, the public key of the signer is needed, which is in the certificate of the signer. This certificate is signed by a trusted authority. The recursion can go on until we arrive at ~~the~~ a root certificate, which is something that we trust because it was distributed through a trusted channel, for example shipped with the web server.

The most secure place to store ~~such~~ a private key is a security token. A security token is a data processing system ~~which~~ that is portable and usable in connection with another data processing sytem or integrated into another data processing system comprising at least a RAM, a ROM, a EEPROM and a microprocessor including specialized functions for accomplishing secure cryptographical methods. A ~~smartcard~~ smart card can be considered as the most convenient and most portable security token ~~even~~ given the current state of the technology. Modern smart cards are able to perform the signing operation inside the card. At the same time they do not provide any function to export the private key to the outside.

During the import of user certificates to a security token (e.g smart card), the validity of a certificate cannot be checked on the security token. This may create errors during the storage of the certificate objects and afterwards during the usage of such stored erroneous certificate on the token. The usage of key and certificate objects stored in the token cannot be ~~guaranteed~~ guaranteed without a valid root certificate of the certification authority (CA) ~~which~~ that generated the user certificate. The root certificate ~~may~~ can only be retrieved from an external database. For example, the user has to search and to retrieve the correct root certificate from an externally available central trusted location (such as an LDAP directory) and after verification of this certificate, extract the public key of the root certificate. This is a very time consuming process.

Furthermore, the external database will prohibit the secure use of the related keys stored on the token for off-line operations.

The user certificate will not be securely stored on a token and thus cannot be trusted by applications using a token for signature generation and verification. The validity of certificates stored on a token cannot be verified completely off-line.

US Patent 5,680,458 deals with a method of replacing a private root key when the private root key has been compromised and the recipient of a signed document can no longer be sure that the document was signed by the certifying authority and not by a party ~~which~~ that compromised the private key. There is no teaching or suggestion in this patent how a user certificate may be securely stored, used or replaced by security tokens.

Objects of the Invention

It is therefore an object of the present invention to provide improved protection of digital certificates and related keys on a security token.

It is a further object of the present invention to provide a secure import of user certificates into a security token.

Finally, it is a further object of the present invention to provide a secure verification of the user certificate stored on a token.

These objects are solved by the features of the independent claims.

Preferred embodiments of the present invention are laid down in the dependent claims.

Summary of the invention

The present invention relates to a system and method for secure usage of digital certificates and related keys on a security token, and more particularly, a secure import of certificates into a security token and their secure usage by applications.

The root certificate of the certification authority(CA) is used during the initialization of the security token in a secure environment to transfer the certified public root key of the CA and its attributes into the data structure of the security token. The public ~~rootkey~~ root key is ~~being~~ write protected in the token. Furthermore, a verification component preferably part of the operating system of the security token will accept afterwards, in a case the certificate has to be replaced, only user certificates having a valid digital signature by the private root key of the CA.

Any application using the user certificates and its related user private keys on the token is able to verify the user certificate using this secure public root key of the CA stored on the token. Preferably, the verification of the user certificate is then even possible during the off-line operation by using the extracted trusted public key of the CA stored on the token.

Brief Description of the Drawings

The present invention will be described in more detail using preferred embodiments with the Figures, where

FIG. 1 shows structure and components of a smart card ~~which may~~ that can be used as a security token;

FIG. 2 shows the content of the EEPROM after initialization of the ~~smartcard~~ smart card according to the present invention;

FIG. 3 shows a flow chart for verification of a new user certificate on the smart card according to the present invention;

FIG. 4 shows a flow chart for creating a signature using the present invention.

Detailed Description of the Preferred Embodiment

A security token ~~may~~ can be used in connection with any portable data processing device, e.g personal digital assistant or mobile phone. The present invention will be described in detail on a smart card ~~which~~ that may be used in a preferred embodiment.

The chip(10) of the smart card (FIG. 1) used by the present embodiment consists of a microprocessor(12), ROM(Read Only Memory; 18), EEPROM(Electrical Erasable Programmable Read Only Memory; 16) and RAM(Random Access Memory; 14). Today, most ~~smartcards~~ smart cards have an 8-bit microprocessor and in the high end cards there are 16- bit or 32-bit ~~processor~~ processors available.

A cryptographical processor as used by the present invention is needed for performing signature operations on the card itself. The user's private key never needs to leave the smart card.

The information stored in the ROM(18) is written during chip manufacturing. It contains the operating system and security algorithms (e.g. DES, RSA).

The EEPROM(16) is used for permanent storage of data and is used ~~as~~ for storage of user certificates, the public key of the CA and root certificate of the CA as well as routines for accomplishing the

present invention, e.g. verification of user certificates. This information will be written into the EEPROM(16) during initialization of the smart card preferably. The RAM(14) is the transient memory of the smart card and keeps the data only as long as the card is powered.

FIG. 2 shows the content of an EEPROM (1) of a smart card presented to carry out the preferred embodiment of the present invention. At manufacturing time, especially during personalization or initialization of the smart card, the root certificate(2) of the certificate authority (CA) and the public root key(4) of the CA extracted from the root certificate(2) are securely stored as objects in the EEPROM(1). Both objects(2,4) are stored via an access condition so that they cannot be replaced or deleted by unauthorized operations after the smart card has been issued. The validity dates contained in the root certificate(2) are used to limit the usage of the smart card and the user's key and certificates. There may be several key pairs and related certificates of one or many ~~user~~ users stored on the smart card. The maximum number of key pairs(n) to be stored in the EEPROM(1) are defined during creation (e.g. personalization) of the smart card.

The object user public key (8) may be stored additionally in the EEPROM of the smart card allowing applications to obtain the public keys of the user faster instead of extracting them from the user certificates. This applies accordingly for the certification authority's public root key (4) ~~which~~ that may be stored additionally in the smart card.

FIG. 3 shows the single steps of the verification routine ~~which~~ that

may be part of the smart card's operating system or may be a separate component called by the operating system or other functions.

A new user key pair (e.g RSA public and private ~~key~~ keys) may be securely generated on the smart card. When the certificate is requested at the CA by the user for one of his public keys, this is done together with the Root Certificate of the CA stored on the smart card. After the CA has tested the information provided by the user and the root certificate of the CA, the CA generates a new user certificate for a new public key.

The new user certificate is returned by the CA to the user's client system and is then stored on the smart card. The smart card operating system validates this new user certificate by checking the digital signature using the stored public root key of CA and the signature algorithm (e.g RSA, ECC, DSA). When the signature is valid, the new user certificate is valid.

The verification routine is called every time a new certificate object has to be stored on the card, especially during the initialization/personalization of the smart card with the user's certificates at card issuing time or during the storage of a replacement certificate at the user's or administrator's client system when e.g the original user certificate has expired. A new user certificate is only accepted by the smart card when the digital signature of the ~~certificate~~ CA provided with the certificate is successfully verified on the card using the public root key of the CA.

The verification routine comprises as least following steps:

1. Sending a new certificate from the CA to a data processing system ~~which~~ that communicates via a wired or wireless connection

with a security token, e.g. ~~smartcard~~ a smart card via starting point(30) in Fig. 3.;

2. Checking the availability of a public root key in the EEPROM of the smart card(40);

3. Storing the new certificate as a temporary object in the EEPROM of the smart card if a public root key is available(50);

4. Generating a HASH over the new user certificate temporarily stored in the ~~smartcard~~ smart card (50);

5. Verifying the digital signature contained in the new user certificate and using the public root key stored in EEPROM for decrypting the digital signature(50);

6. Comparing the HASH generated by step 4 with the HASH generated by step 5; if both are identical then the new certificate is authenticated (60);

7. Creating a new user certificate object on the smart card and deleting or validating the temporary user certificate (80);

8. Optionally, to improve the linking of the user public key, user private key, and user certificate for the public key, these three objects are available as a group with the same ID via the application interface for creation and verification of digital signatures.

The new user certificate consists of two parts. The first part, for example, contains data elements relating to the key, the issuer of the certificate, the user, the signature algorithm, the serial number, etc. The second part of the certificate contains a digital signature relating to the first part of the certificate. A digital

signature basically establishes the authenticity of electronically transmitted messages or electronic documents. The process of generating a digital signature can be presented as follows.

From the first part of the certificate a HASH algorithm(e.g.SHA- 1, MD5) is used to form a HASH value. The HASH algorithm compresses the data from the first part of the certificate. Then the HASH value is decrypted with a crypto algorithm. Decryption is based on the private key of a key pair. In the present case the new certificate is encrypted with the private key of the CA.

FIG. 4 shows the communication between the smart card and an application installed on a data processing system using the present invention.

At a first time a communication is established between an application running on a data processing system and a smart card, the verification routine verifies the availability of the Root Certificate of a CA on the smart card (110). Then, the application obtains the certificate from the smart card, verifies the standard information stored in the certificate (e.g expiration date), retrieves the public root key from the certificate (110) and gets a selected user certificate from the smart card ~~which~~ that will be used for creating a digital signature. Before that user certificate may be used, the verification routine verifies the digital signature contained in that user certificate, generates a HASH using the HASH algorithm specified in the user certificate and uses the public root key for decrypting the digital signature attached to the user certificate. If both HASHs are identical then the user certificate is authenticated (130).

Finally, a HASH is generated over the message to be signed, the HASH is encrypted with the private key and signature algorithm specified in the user certificate, resulting in a digital signature

(150). The digital signature is attached to the message to be sent(170). A correctly signed message has been generated with the correct user certificate, which proves the validity and the authenticity of the message when received via an insecure network~~(180)~~.

The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods described herein, and which - when loaded in a computer system - is able to carry out these methods.

A B S T R A C T

The present invention relates to a security token and method for secure usage of digital certificates and related keys on a security token, and more particularly, a secure import of certificates into a security token and their secure usage by applications. The root certificate of the certification authority(CA) is used during the initialization of the security token in a secure environment to transfer the certified root public key of the CA and its attributes into the data structure of the security token. The public root key is write protected. Furthermore, a verification component, preferably part of the operating system of the security token will accept, ~~in case~~ in case the certificate has to be replaced, only user certificates having a valid digital signature by the private root key of the CA.